Adaptive representation for single objective optimization

1 Introduction

Adaptive Representation Evolutionary Algorithm (AREA) is similar to the Evolution Strategy (ES) technique [11, 12] as it uses a population of individuals which are modified by mutation. Whereas the ES individuals have a fixed representation (binary or real), the AREA individuals use an adaptive representation that may be changed during (and without halting) the search process.

ES employs a special mechanism for adapting the mutation parameter. For instance standard ES tries to adapt the standard deviation parameter when the Gaussian perturbation is used. These adaptations are of very little help. It is so because the function to be optimized is usually very intricate and the optimal parameter setting for a certain region of the search space may not be optimal for a neighboring region, at least.

Moreover, an incorrect setting for the value of the mutation parameter may lead to poor results. For instance, if the mutations are rare, the population could (and often will) converge to a local optimal point. If the mutations occur too often, the evolutionary process has a random search character. Facing these two problems, AREA employs a new way of searching through the solution space.

Many examples from nature can be found in order to sustain AREA. The first example resides in the human DNA which is, roughly speaking, a string of nucleotides over the alphabet \{T (thymine), C (cytosine), G (guanine), A (adenine)\} [6, 9]. By contrast, the standard evolutionary algorithms use strings over the alphabet \{0, 1\} which consists of only two values. The four nucleotide system has been developed under the specific conditions of the Earth environment. Had different conditions been on Earth, maybe a ten nucleotide system could have developed. Lately, the entire evolution was based on this alphabet made up of only four symbols.

If we take a look at the history of the Earth we can see that the species evolved very slowly. Billions of years were needed to develop the diversity and perfection of life we know today. The entire evolution (of the complex structure) is based on reproduction by recombination and mutation. Recombination ensures the perpetuation of life. Mutation is responsible for maintaining diversity and for exploring new functional ways to combine the nucleotides.

If only two nucleotide systems had been used, the length of the human DNA (that encodes the same diversity) would probably have been very large. A mutation on this chromosome would probably be too small to produce a significant change. But, too many mutations would produce dramatic changes and the obtained individual would not survive.
If ten nucleotide systems had been used, the length of the human DNA (that encodes the same diversity) would have (probably) been very small. A mutation on this chromosome would have produced a significant change and the species diversity would have been greater.

AREA is essentially a technique that works with higher alphabets. Each AREA individual consists of a pair \((x, B)\) where \(x\) is a string encoding object variables and \(B\) specifies the alphabet used for encoding \(x\). Binary encoded strings are a particular case of AREA.

If only one alphabet had been used the gain of AREA over standard ES would have been minimal. Thus, the AREA individuals use a dynamic system of alphabets that may be changed during (and without halting) the search process. If an individual gets stuck in a local optimum—from where it is not able to “jump”—, the individual representation is changed, hoping that this new representation will help the individual to escape from the current position and to explore farther and more efficiently the search space.

The similarities between the AREA behavior and the behavior of other species from nature are also numerous. For instance, the chameleon which is able to change the color of its skin depending on the place. The AREA individuals possess the same ability to change their looks as the chameleon. From this point of view AREA may be considered as an interesting case of chameleonic programming.

Taking into account the No Free Lunch Theorems [16] (NFL) we cannot say that AREA is better than other evolutionary algorithms for all of the test problems. Indeed, several cases where other evolutionary algorithms used for comparison are better than AREA have been successfully identified. However, AREA significantly outperforms the standard evolutionary algorithms on the well-known difficult (multimodal) test functions. This advantage of AREA makes it very suitable for real-world applications where we have to deal with highly multi-modal functions.

Like GP and GA, AREA is also subject to a debate concerning the benefit of the genome-phenome systems over the genome only systems. As stated in many papers there is a question whether the maintaining of multiple, different genomes that encode the same phenotype should be beneficial. AREA maintains multiple different genomes (strings encoded over different alphabets) that encode the same phenotype (points in the search space) and this ability seems to be very beneficial.

AREA relies mainly on Dynamic Representation (DR) proposed in [10]. Both AREA and DR use higher alphabets for encoding solutions and a special mutation operator that changes the encoding alphabet during the search process. Whereas DR alphabets changing are blind, AREA employs an efficient strategy for changing the encoding alphabets.

The problems of adapting individual representation and the parameters of an evolutionary algorithm are difficult. They have been studied since the birth of genetic algorithms and evolutionary strategies. Some aspects of that study are described in [1–5], [13–15].

The paper is organized as follows. The AREA technique is described in Sect. 2. The algorithms used in experiments are presented in Sect. 3. Several numerical experiments using are performed in Sect. 4. The test functions involved in the numerical experiments are well-known benchmarking problems used to assess the performances of the evolutionary algorithms. Most of these functions are highly multimodal employing different difficulties of the search space. Several important issues regarding the AREA representation are discussed in Sect. 5.

2 The AREA technique

The main idea of this technique is to allow each solution be encoded over a different alphabet. Moreover, the representation of a particular solution is not fixed. Solution representation is adaptive and may be changed during the search process as an effect of the mutation operator.

2.1 Solution representation

Each AREA individual consists of a pair \((x, B)\) where \(x\) is a string encoding object variables and \(B\) specifies the alphabet used for encoding \(x\). \(B\) is an integer number, \(B \geq 2\), and \(x\) is a string of symbols from the alphabet \(\{0, 1, \ldots, B - 1\}\). If \(B = 2\), the standard binary encoding is obtained.

Each solution has its own encoding alphabet. The alphabet over which \(x\) is encoded may change during the search process.

When no ambiguity arises we will use \(B\) to denote the alphabet \(B = \{0, 1, \ldots, B - 1\}\).

An example of an AREA chromosome is the following:

\[
C = (301453, 6).
\]

Remark. The genes of \(x\) may be separated by comma if required. For instance the comma separator is always needed when \(B \geq 10\).

2.2 Mutation

Mutation can modify object variables as well as the last position (specifying the representation alphabet).

When the changing gene belongs to the object variable substring (\(x\) - part of the chromosome) the mutated gene is a randomly chosen symbol from the same alphabet.

Consider the chromosome \(C\) represented over the alphabet \(B = 8\):

\[
C = (631751, 8).
\]